Part 1 : Basic data types

Code	Console

```let name = "Oussama";```
```console.log(name)```
```console.log(typeof name)```	

```let age = 30;```
```let is_student = true;```
```console.log(age, is_student);```	

```let sum = 10 + 5;```
```let product = 10 * 5;```
```let isEqual = (10 === 10);```
```console.log(sum, product, isEqual);```	

```let var = true;``` ```console.log(name +" false")```	

```let name = true;```
```console.log(name + 2)```
```console.log(name / 2)```	

```let age = 20;```
```if (age >= 18) {```
```   console.log("Adult");```
```} else {```
```   console.log("Not an adult");```
```}```	

```const age = 26;```
```const beverage = age >= 21 ? "milk" : "Juice";```
```console.log(beverage);```	

```let x = 0.1;```
```let y = 0.2;```
```let z = x + y;```
```console.log(z);```	

```let a = "10";```
```let b = 20;```
```let result = a + b;```
```console.log(result);```	

```let num1 = 10;```	

```let num2 = "10";```
```console.log(num1 == num2);```
```console.log(num1 === num2);```	

```let str = "5";```
```let num = 2;```
```let result = str - num;```
```console.log(result);```	

```let null_value = null;```
```let undefined_value;```
```console.log(null_value);```
```console.log(undefined_value);```	

```let is_raining = true;```
```if (is_raining) {```
```   console.log("Take an umbrella.");```
```} else {```
```   console.log("Enjoy the sunshine.");```
```}```	

```let temperature = 20, isSunny = true;```
```if (temperature > 15) {```
```   if (isSunny) {```
```   console.log("It's a beautiful day!");```
```   } else {```
```   console.log("It's cloudy, but warm.");```
```   }```
```} else {```
```   console.log("It's cold outside.");```
```}```	

```let dayOfWeek = "Tuesday";```
```switch (dayOfWeek) {```
```   case "Monday":```
```   console.log("It's the start of the week.");```
```   break;```
```   case "Tuesday":```
```   case "Wednesday":```
```   case "Thursday":```
```   case "Friday":```
```   console.log("It's a weekday.");```
```   break;```
```   case "Saturday":```
```   case "Sunday":```
``` console.log("It's the weekend!");```	

<table>
<tr><td>

```
    break;
  default:
    console.log("Invalid day.");
}
```

</td><td></td></tr>
<tr><td>

```
let isStudent = true;
let hasDiscount = false;

if (isStudent && hasDiscount) {
   console.log("You qualify for a student discount.");
} else if (isStudent || hasDiscount) {
   console.log("You qualify for a discount.");
} else {
   console.log("No discount available.");
}
```

</td><td></td></tr>
</table>

Homework :

1. Write a program that takes a temperature value in Celsius and determines whether it's considered cold (below 15) , warm (between 15 and 25) , or hot (above 25).

2. **Complete the table below**

Code	Console

```let sum = 9 , str = "0";;```
 	
 ```console.log(sum% 3 === 0+0)```
 	
 ```console.log(sum% 3 === 0+"0")```
 	
 ```console.log(sum% 3 == 0+"0")```
 	
 ```console.log(sum % 3 === 0 + 0)```
 	
 ```console.log(sum % 3 === 0 + str)```
 	
 ```console.log(sum % 3 == 0 + str)```
 	
 ```console.log(number + str)```
 	
 ```console.log(sum + +str)```
 	
 ```let boolVal = true```
 	
 ```console.log(number % 3 === boolVal)```	

Part 2 : Loops

Code	Console

``js
let sum = 0;
 	
for (let i = 1; i <= 10; i++) {
  if (i === 3) continue;
   sum += i;
}
console.log(sum);
``	

``js
let sum = 0;
for (let i = 1; i <= 10; i++) {
   sum += i;
}
console.log("Sum of numbers from 1 to 10:", sum);
``	

``js
let number = 5;
let factorial = 1;
 	
for (let i = 1; i <= number; i++) {
   factorial *= i;
}
console.log("Factorial of 5:", factorial);
``	

``js
let number = 7;
for (let i = 1; i <= 3; i++) {
   console.log(`${number} x ${i} = ${number * i}`);
}
``	

``js
let number = 29;
let isPrime = true;
for (let i = 2; i <= Math.sqrt(number); i++) {
   if (number % i === 0) {
     isPrime = false;
     break;
   }
}
console.log(`${number} is prime:`, isPrime);
``	

Part 3 : Arrays

Code	Console
let fruits = ['apple', 'banana', 'cherry']; console.log("Fruits array:", fruits);	

let numbers = [1, 2, 3];
numbers.push(4);
console.log(**typeof** numbers)
console.log(numbers);	

let numbers = [1, 2, 3, 4];
numbers.pop();
console.log(numbers);	

let numbers = [1, 2, 3, 4, 5]; console.log("Length of the array:", numbers.length);	

let numbers = [1, 2, 3, 4, 5];
let doubledNumbers = numbers.map(num => num * 2);
console.log(doubledNumbers);	

let numbers = [1, 2, 3, 4, 5];
let evenNumbers = numbers.filter(num => num % 2 === 0);
console.log(evenNumbers);	

let fruits = ['apple', 'banana', 'cherry'];
let hasBanana = fruits.includes('banana');
console.log(hasBanana);	

a=[1,2,3] b=a a.pop() console.log(b)	
a=[1,2,3] b=[...a] a.pop() console.log(b)	

let arr1 = [1, 2, 3];
let arr2 = [4, 5, 6];
let combinedArray = [...arr1, ...arr2];
console.log(combinedArray);	

let numbers = [4, 2, 9, 1, 5];
numbers.sort((a, b) => a - b);
console.log(numbers);	

```javascript
let fruits = ['apple', 'banana', 'cherry'];
let index = fruits.indexOf('cherry');
console.log("Index of 'cherry':", index);
let index_two = fruits.indexOf(orange);
console.log("Index of orange:", index_two );
```

```javascript
// HomeWork
let people = [
    { name: "Ahmed", age: 28, hobbies: ["reading", "hiking"] },
    { name: "Ibn Taymiya", age: 34, hobbies: ["cooking", "cycling"] },
    { name: "Ibn Hibane", age: 22, hobbies: ["running"] },
    { name: "Abou hanifa", age: 40, hobbies: ["reading", "travelling"]
},
    { name: "Khadija", age: 31, hobbies: ["photography", "gaming"] }
];

let overThirty = people.filter(person => person.age > 30);
console.log("People over 30:", overThirty);

let allHobbies = [...new Set(people.flatMap(person =>
person.hobbies))];
console.log("All unique hobbies:", allHobbies);

let hobbyCounts = allHobbies.map(hobby => {
    return {
        hobby: hobby,
        count: people.filter(person =>
person.hobbies.includes(hobby)).length
    };
});
console.log("Hobby counts:", hobbyCounts);

let multiHobbyists = people.filter(person => person.hobbies.length >
1);
let averageAge = multiHobbyists.reduce((sum, person) => sum +
person.age, 0) / multiHobbyists.length;
console.log("Average age of people with more than one hobby:",
averageAge);


let peopleList = people.map(person => `${person.name}
(${person.age})`).join(", ");
console.log("List of people and ages:", peopleList);
```

Part 4 : Strings

Code	Console

```let text= "Hello, World!";```
```console.log(text);```
```console.log("Length of the string:", text.length);```
```console.log("First character:", text[0]);```
```console.log("Last character:", text[text.length - 1]);```
```console.log("Uppercase:", text.toUpperCase());```
```console.log("Lowercase:", text.toLowerCase());```	

```let firstName = "Mohamed";```
```let lastName = "Ibn Abd Elouahab";```
```let fullName = firstName + " " + lastName;```
```console.log("Full name:", fullName);```	

```let name = "Alice";```
```let message = `My name is ${name}`;```
```console.log(message);```	

```let text = "My name is A'icha";```
```let position = text.indexOf("name");```
```console.log("Position of 'fox':", position);```	

```let text = "JavaScript Programming";```
```let substring = text.substring(0, 10);```
```console.log(substring);```	

```let sentence = "This is a sentence.";```
```let words = sentence.split(" ");```
```console.log("Words array:", words);```	

```let text = "I love JavaScript!";```
```let newText = text.replace("JavaScript", "Python");```
```console.log("Replaced text:", newText);```	

```let text = "Hello, World!";```
```let containsHello = text.includes("Hello");```
```console.log("Contains 'Hello':", containsHello);```	

```let text = "   JavaScript   ";```
```let trimmedText = text.trim();```
```console.log("Trimmed text:", `'${trimmedText}'`);```	

```let number = 123;```
```let numberString = number.toString();```
```console.log("Number as a string:", numberString);```	

let number = 123;
let numberString = number.toString();
console.log("Number as a string:", numberString);	

let text = "JavaScript";
let reversedText = text.split("").reverse().join("");
console.log("Reversed string:", reversedText);	

let text = "JavaScript"; console.log("Character at index 4:", text.charAt(4));	

let text1 = "Hello";
let text2 = "World";
let result = text1.concat(", ", text2, "!");
console.log("Concatenated string:", result);	

let text = "Hello World!";
console.log("Ends with 'World!':",
text.endsWith("World!"));
 	
console.log("Ends with 'Hello':",
text.endsWith("Hello"));	

let text = "JavaScript is fun";
console.log("Index of 'is':", text.indexOf("is"));
console.log("Index of 'Java':", text.indexOf("Java"));	

let text = "JavaScript is fun and JavaScript is powerful"; console.log("Last index of 'JavaScript':", text.lastIndexOf("JavaScript"));	

let text = "The rain in SPAIN stays mainly in the plain";
let result = text.match(/ain/g);
console.log("Matched substrings:", result);	

let text = "Hello, World!";
let result = text.slice(0, 5);
console.log("Sliced string:", result);	

let text = "Hello World!"; console.log("Starts with 'Hello':", text.startsWith("Hello")); console.log("Starts with 'World':", text.startsWith("World"));	

Strings Homework

Code

```javascript
let input = "Hello123World!456JavaScript789";

let numbers = input.match(/\d+/g);
let sum = numbers.reduce((acc, num) => acc + parseInt(num), 0);
console.log("Sum of all numbers in the string:", sum);

let alphabets = input.match(/[a-zA-Z]/g).reverse();
let reversedString = input.replace(/[a-zA-Z]/g, () => alphabets.shift());
console.log("String with reversed letters:", reversedString);

let binaryString = input.replace(/\d+/g, (match) => parseInt(match).toString(2));
console.log("String with numbers replaced by binary:", binaryString);

let swappedCaseString = input.replace(/[a-zA-Z]/g, (char) =>
    char === char.toUpperCase() ? char.toLowerCase() : char.toUpperCase()
);
console.log("String with swapped case:", swappedCaseString);
```

Console

Part 5 : Objects

Code	Console

``javascript
let person = {
    name: "Alice",
    age: 30,
    job: "Engineer"
};
console.log(person.name);
console.log(person['age']);
 	
person.country = "USA";
console.log(person.country);
``	

``javascript
let company = {
    name: "TechCorp",
    address: {
        street: "123 Main St",
        city: "Techville",
        zip: "12345"
    },
    employees: [
        { name: "John", role: "Developer" },
        { name: "Jane", role: "Designer" }
    ]
};
 	
console.log(company.address.city);
console.log(company.employees[1].name);
``	

``javascript
let calculator = {
    number: 0,
 	
    add: function(value) {
        this.number += value;
        return this.number;
    },
 	
    subtract: function(value) {
        this.number -= value;
        return this.number;
    }
};
 	
console.log(calculator.add(10));
console.log(calculator.subtract(4));
``	

``javascript
let car = {
    make: "Toyota",
    model: "Corolla",
    year: 2020
};
 	
for (let key in car) {
    console.log(key + ": " + car[key]);
}
``	

``javascript
let student = {
    name: "Emily",
    age: 22,
    major: "Physics"
};
let { name, age, major } = student;
console.log(name);
console.log(age);
console.log(major);
``	

``javascript
let original = { a: 1, b: 2 };
let clone = { ...original };
console.log(clone);
 	
let obj1 = { x: 10, y: 20 };
let obj2 = { y: 30, z: 40 };
let merged = { ...obj1, ...obj2 };
console.log(merged);
``	

``javascript
let book = {
    title: "1984",
    author: "George Orwell",
    year: 1949
};
console.log(Object.keys(book));
console.log(Object.values(book));
console.log(Object.entries(book));
``	

``javascript
let counter = {
    value: 0,
    increment() {
        this.value++;
        console.log(this.value);
    }
};
counter.increment();
counter.increment();
``	

```
let person = {
    firstName: "John",
    lastName: "Doe",
    get fullName() {
        return `${this.firstName} ${this.lastName}`;
    },
    set fullName(name) {
        let parts = name.split(" ");
        this.firstName = parts[0];
        this.lastName = parts[1];
    }
};
console.log(person.fullName);
person.fullName = "Jane Smith";
console.log(person.firstName);
console.log(person.lastName);
```

Home Work

1. You have a data structure representing a company with multiple departments. Each department has teams, and each team has employees.

```
let company = {
    name: "TechCorp",
    departments: {
        engineering: {
            teams: {
                frontend: [
                    { name: "Aissa", salary: 90000 },
                    { name: "Salah", salary: 85000 }
                ],
                backend: [
                    { name: "Houde", salary: 95000 },
                    { name: "Nouh", salary: 90000 }
                ]
            }
        },
        marketing: {
            teams: {
                digital: [
                    { name: "Adam", salary: 70000 },
                    { name: "Abbass", salary: 75000 }
                ],
                content: [
                    { name: "Oussama", salary: 80000 },
                    { name: "Ibn Taymiya", salary: 85000 }
                ]
            }
```

```
    },
    sales: {
      teams: {
        domestic: [
          { name: "Ayoub", salary: 65000 },
          { name: "Younes", salary: 72000 }
        ],
        international: [
          { name: "Souhaib", salary: 78000 },
          { name: "Moussa", salary: 83000 }
        ]
      }
    }
  }
};
```

The task is to perform the following operations:

1. Calculate the total number of employees in the company.

2. Find the average salary of employees across all departments.

3. List the names of all employees who earn above a certain threshold.

4. Update the salary of a specific employee given their name and department

2. you have a nested object representing a school with multiple classes, each class containing students.

```
let school = {
  classes: {
    classA: {
      students: [
        { name: "Aymen", scores: { math: 85, english: 92, science: 88 } },
        { name: "Mohamed", scores: { math: 78, english: 81, science: 85 } }
      ]
    },
    classB: {
      students: [
        { name: "Ahmed", scores: { math: 90, english: 85, science: 88 } },
        { name: "Ayoub", scores: { math: 88, english: 89, science: 90 } }
      ]
    },
    classC: {
      students: [
        { name: "Souhaib", scores: { math: 92, english: 91, science: 89 } },
        { name: "Youcef", scores: { math: 80, english: 84, science: 85 } }
      ]
    }
  }
};
```

The tasks are:

1. Find the top-performing student in each class based on their scores.

2. Calculate the average score for each class and the overall average score.

3. List all students who have scored above a certain threshold in any of their subjects.

4. Update the score of a specific student in a given subject.

Part 6 : Functions

Code	Console

``js
function add(a, b) {
    return a + b;
}
console.log(add(5, 3));
``	

``js
function greet(name) {
    return `Hello, ${name}!`;
}
console.log(greet("Ahmed"));
``	

``js
function findMax(x, y) {
    return x > y ? x : y;
}
console.log(findMax(10, 20));
``	

``js
function isEven(num) {
    return num % 2 === 0;
}
console.log(isEven(10));
console.log(isEven(7));
``	

``js
function factorial(n) {
    return n <= 1 ? 1 : n * factorial(n - 1);
}
console.log(factorial(5));
``	

``js
function isPalindrome(str) {
    const reversed = str.split("").reverse().join("");
    return str === reversed;
}
console.log(isPalindrome("racecar"));
console.log(isPalindrome("hello"));
``	

``js
function filterEvens(arr) {
    return arr.filter(num => num % 2 === 0);
}
console.log(filterEvens([1, 2, 3, 4, 5, 6]));
``	

``js
function makeMultiplier(multiplier) {
    return function(num) {
        return num * multiplier;
    };
}
const double = makeMultiplier(2);
console.log(double(5));
``	

| ```javascript
function flattenArray(arr) {
   return arr.reduce((flat, toFlatten) =>
      flat.concat(Array.isArray(toFlatten) ? flattenArray(toFlatten) :
toFlatten), []);
}
console.log(flattenArray([1, [2, [3, 4]], 5]));
``` | |
| ```javascript
function getRandomInRange(min, max) {
   return Math.floor(Math.random() * (max - min + 1)) + min;
}
console.log(getRandomInRange(1, 10));
``` | |
| ```javascript
const isOdd = num => num % 2 !== 0;
console.log(isOdd(7));
console.log(isOdd(10));
``` | |
| ```javascript
const add = (a, b) => a + b;
console.log(add(5, 3));
``` | |
| ```javascript
const square = x => x * x;
console.log(square(4));
``` | |
| ```javascript
const filterEvens = arr => arr.filter(num => num % 2 === 0);
console.log(filterEvens([1, 2, 3, 4, 5, 6]));
``` | |
| ```javascript
function sumArray(arr) {
   if (arr.length === 0) return 0;
   return arr[0] + sumArray(arr.slice(1));
}
console.log(sumArray([1, 2, 3, 4, 5]));
``` | |
| ```javascript
function countDown(num) {
   if (num < 0) return;
   console.log(num);
   countDown(num - 1);
}
countDown(5);
``` | |
| ```javascript
function power(base, exponent) {
   if (exponent === 0) return 1;
   return base * power(base, exponent - 1);
}
console.log(power(2, 3));
``` | |
| ```javascript
function reverseString(str) {
   if (str === "") return "";
   return reverseString(str.substr(1)) + str.charAt(0);
}
console.log(reverseString("hello"));
``` | |

```javascript
function reverseString(str) {
    if (str === "") return "";
    return reverseString(str.substr(3)) + str.charAt(0);
}
console.log(reverseString("hello"));
```

```javascript
// Home Work
function tower_of_hanoi(n, source, target, auxiliary) {
    if (n === 1) {
        console.log(`Move disk 1 from ${source} to ${target}`);
        return;
    }
    tower_of_hanoi(n - 1, source, auxiliary, target);
    console.log(`Move disk ${n} from ${source} to ${target}`);
    tower_of_hanoi(n - 1, auxiliary, target, source);
}
tower_of_hanoi(3, 'A', 'C', 'B');
```

```javascript
// Home Work
function permute(str, l = 0, r = str.length - 1) {
    if (l === r) {
        console.log(str);
    } else {
        for (let i = l; i <= r; i++) {
            str = swap(str, l, i);
            permute(str, l + 1, r);
            str = swap(str, l, i); // backtrack
        }
    }
}
function swap(s, i, j) {
    let charArray = s.split('');
    [charArray[i], charArray[j]] = [charArray[j], charArray[i]];
    return charArray.join('');
}
permute("abc");
```

```javascript
// Home Work
function subsets(set) {
    const result = [];
    const total = 1 << set.length; // 2^n

    for (let i = 0; i < total; i++) {
        const subset = [];
        for (let j = 0; j < set.length; j++) {
            if (i & (1 << j)) {
                subset.push(set[j]);
            }
```

```
        }
        result.push(subset);
    }
    return result;
}
console.log(subsets([1, 2, 3]));
```

```
function gcd(a, b) {
    if (b === 0) return a;
    return gcd(b, a % b);
}
console.log(gcd(48, 18));
```

```
const numbers = [1, 2, 3, 4, 5];
const sum = numbers.reduce((accumulator, currentValue) => {
    return accumulator + currentValue;
}, 0); // Initial value is 0
console.log(sum);
```

```
function get_ighest_paid_pmployee(employees) {
    if (employees.length === 0) return null;

    return employees.reduce((highest, current) => {
        return (current.salary > highest.salary) ? current : highest;
    });
}

const employees = [
    { id: 1, name: "Alice", salary: 70000 },
    { id: 2, name: "Bob", salary: 85000 },
    { id: 3, name: "Charlie", salary: 60000 },
];

const highestPaid = get_ighest_paid_pmployee(employees);
console.log("Highest Paid Employee:", highestPaid);
```

Part 7 : OOP

Code	Console
```js\nclass Employee {\n constructor(name, salary) {\n this.name = name;\n this.salary = salary;\n }\n\n getDetails() {\n return `${this.name} earns $$${this.salary}`;\n }\n}\nconst emp1 = new Employee("Alice", 50000);\n\nconsole.log(emp1.getDetails());\n```	
```js\nclass Employee {\n constructor(name, salary) {\n this.name = name;\n this.salary = salary;\n }\n\n getDetails() {\n return `${this.name} earns $$${this.salary}`;\n }\n}\n\nclass Manager extends Employee {\n constructor(name, salary, department) {\n super(name, salary);\n this.department = department;\n }\n\n getDetails() {\n return `${super.getDetails()} and manages the ${this.department} department.`;\n }\n}\n\nconst mgr1 = new Manager("Bob", 80000, "Sales");\n\n\nconsole.log(mgr1.getDetails());\n```	

HomeWork

Code

```
class Product {
    constructor(id, name, price, stock) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.stock = stock;
    }
    isAvailable(quantity) {
        return this.stock >= quantity;
    }
    reduceStock(quantity) {
        this.stock -= quantity;
    }
}
class ShoppingCart {
    constructor() {
        this.items = [];
    }
    addProduct(product, quantity) {
        if (product.isAvailable(quantity)) {
            this.items.push({ product, quantity });
            product.reduceStock(quantity);
            console.log(`Added ${quantity} of ${product.name} to the cart.`);
        } else {
            console.log(`Sorry, not enough stock for ${product.name}.`);
        }
    }
    calculateTotal() {
        return this.items.reduce((total, item) => total + item.product.price * item.quantity, 0);
    }
    displayCart() {
        console.log("Shopping Cart:");
        this.items.forEach(item => {
            console.log(`${item.quantity} x ${item.product.name} - $${item.product.price} each`);
        });
        console.log(`Total: $${this.calculateTotal()}`);
    }
}
class Order {
    constructor(cart) {
        this.cart = cart;
        this.orderDate = new Date();
    }
```

```
  processOrder() {
     console.log("Processing Order...");
     this.cart.displayCart();
     console.log("Order has been placed successfully!");
  }
}
const shirt = new Product(1, "T-Shirt", 19.99, 10);
const jeans = new Product(2, "Jeans", 39.99, 5);
const jacket = new Product(3, "Jacket", 59.99, 2);

const cart = new ShoppingCart();
cart.addProduct(shirt, 2);
cart.addProduct(jeans, 1);
cart.addProduct(jacket, 3);

const order = new Order(cart);
order.processOrder();
```

Console

Part 8 : Dates

Code	Console
const now = new Date(); console.log(now);	
const specificDate = new Date('2024-01-01'); console.log(specificDate);	
const currentYear = new Date().getFullYear(); console.log(currentYear);	
const currentMonth = new Date().getMonth() + 1; console.log(currentMonth);	
const currentDay = new Date().getDate(); console.log(currentDay);	
const dayOfWeek = new Date().getDay(); console.log(dayOfWeek);	
const formattedDate = new Date().toDateString(); console.log(formattedDate);	
const timeInMilliseconds = new Date().getTime(); console.log(timeInMilliseconds);	

const date = new Date();
date.setFullYear(2025);
console.log(date);	

const today = new Date();
const daysToAdd = 5;
today.setDate(today.getDate() + daysToAdd);
console.log(today);	

const date1 = new Date('2024-01-01');
const date2 = new Date('2024-12-31');
console.log(date1 < date2);	

const startDate = new Date('2024-01-01');
const endDate = new Date('2024-11-31');
const differenceInTime = endDate - startDate;
const differenceInDays = differenceInTime / (1000 * 3600 * 24);
console.log(differenceInDays);	

const isoString = new Date().toISOString(); console.log(isoString);	

const localTimeString = new Date().toLocaleTimeString(); console.log(localTimeString);	
const parsedDate = Date.parse('2024-01-01T00:00:00Z'); console.log(new Date(parsedDate));	

HomeWork

```javascript
class Event {
  constructor(name, startDate, endDate) {
    this.name = name;
    this.startDate = new Date(startDate);
    this.endDate = new Date(endDate);
  }

  conflictsWith(otherEvent) {
    return this.startDate < otherEvent.endDate && this.endDate > otherEvent.startDate;
  }


 formatEventDetails() {
  const options =    { year: 'numeric', month: 'long',
                     day: 'numeric', hour: '2-digit', minute: '2-digit', timeZoneName: 'short' };
    const start = this.startDate.toLocaleString('en-US', options);
    const end = this.endDate.toLocaleString('en-US', options);
    return `${this.name}: ${start} - ${end}`;
  }
}
function scheduleEvent(events, newEvent) {
  for (const event of events) {
    if (newEvent.conflictsWith(event)) {
      console.log(`Conflict detected: ${newEvent.name} conflicts with ${event.name}`);
      return false;
    }
  }
  events.push(newEvent);
  console.log(`Event scheduled: ${newEvent.formatEventDetails()}`);
  return true;
}
const events = [];

const event1 = new Event("Team Meeting", "2024-09-05T10:00:00-05:00", "2024-09-05T11:00:00-05:00");
const event2 = new Event("Project Presentation", "2024-09-05T11:30:00-05:00", "2024-09-05T12:30:00-05:00");
const event3 = new Event("Lunch with Client", "2024-09-05T10:30:00-05:00", "2024-09-05T11:30:00-05:00");
scheduleEvent(events, event1);
scheduleEvent(events, event2);
scheduleEvent(events, event3); // This will conflict with event1
```

Global Homework

You are given a string s consisting of lowercase English letters, and an integer k.

- First, convert s into an integer by replacing each letter with its position in the alphabet (i.e., replace 'a' with 1, 'b' with 2, ..., 'z' with 26).
- Then, transform the integer by replacing it with the sum of its digits.
- Repeat the transform operation k times in total.
- For example, if s = "zbax" and **k = 2**, then the resulting integer would be 8 by the following operations:

  1. Convert: "zbax" → "(26)(2)(1)(24)" → "262124" → 262124
  2. Transform **#1**: 262124 → 2 + 6 + 2 + 1 + 2 + 4 → 17
  3. Transform **#2**: 17 → 1 + 7 → 8
  4. Return the resulting integer after performing the operations described above.

Example 1:

```
Input: s = "iiii", k = 1
Output: 36
Explanation: The operations are as follows:
- Convert: "iiii" → "(9)(9)(9)(9)" → "9999" → 9999
- Transform #1: 9999 → 9 + 9 + 9 + 9 → 36
Thus the resulting integer is 36.
```

Example 2:

```
Input: s = "leetcode", k = 2
Output: 6
Explanation: The operations are as follows:
- Convert: "leetcode" → "(12)(5)(5)(20)(3)(15)(4)(5)" →
"12552031545" → 12552031545
- Transform #1: 12552031545 → 1 + 2 + 5 + 5 + 2 + 0 + 3 + 1 + 5 +
4 + 5 → 33
- Transform #2: 33 → 3 + 3 → 6
Thus the resulting integer is 6.
```